NIC 17161
RFC# 525
Received at NIC June 25, 1973

Postel

COMPUTER SYSTEMS
LABORATORY--UCSB
William Parrish
John R. Pickens
1 June 1973

MIT-MATHLAB MEETS UCSB-OLS
An Example of Resource Sharing

RECEIVED
JUN 28 1973

I. INTRODUCTION

A. Resource Sharing, A Comment

Non-trivial resource sharing among dissimilar systems is a much discussed concept which, to date, has seen only a few real applications. [See NIC 13538, "1972 Summary of Research Activities (UTAH) for a description of Tony Hearn's TENEX-CCN Programming Link.] The first attempts have utilized the most easily accessible communication paths, (TELNET and RJS) and the most universal representations of numbers (byte-oriented numeric characters in scientific notation). Future schemes will probably be more efficient through standardized data and control protocols, but even with the existing approaches users are gaining experience with combinations of resources previously not available.

B. The MATHLAB/UCSB-OLS Experiment

MATHLAB* and OLS are powerful mathematics systems which cover essentially non-intersecting areas of mathematical endeavor. MATHLAB (or MACSYMA) contains a high-powered symbolic manipulation system. OLS is a highly interactive numeric and graphics system which, through user programs, allows rapid formulation and evaluation of problem solutions. Prior to this experiment, users have dealt with problems symbolically on MATHLAB or numerically and graphically on OLS. Lacking an interconnecting data path, users have been left to pencil and paper translation between the two systems.

The goal of the MATHLAB-OLS experiment is to provide an automated path whereby expressions at MATHLAB may be translated into User Programs at UCSB. Thus the user is able to experiment freely with the numeric, graphic, and symbolic aspects of mathematic problems.

---

*Supported on a PDP-10 System at MIT and available for use at UCSB by way of the ARPA Network.

II. THE RESOURCES

To understand this particular case of resource sharing, it is first necessary to understand, to some degree, the resources being shared. This paper does not attempt to deal with all of the resources available at both sites (UCSB and MIT). Only the applicable shared resources are discussed briefly. In the section discussing possibilities for additions (Section V) some available unshared resources are presented, along with their possible shared applications. The current implementation is limited to evaluation of real functions. A description of the capabilities at the two sites follows.

A. Graphical and Numeric Computation Capabilities at UCSB

To get a graph of a function on the OLS, it is necessary only to specify the function with a series of button-pushes. For example, to get a plot on sin(x), the "program"

II REAL SIN x DISPLAY RETURN

will display a plot of sin(x) versus X, provided that X has been defined as a vector containing values over the range which it is desired to plot. For a more complete description of OLS see NIC 5748, "The OLS User's Manual". Programs in OLS, or sequences of button-pushes can be stored under USER level keys, i.e. the above program could be defined as USER LI (+), and the user could display, modify, and look at various values of the SIN function over different ranges by simply setting up the desired value of the vector X, and then typing USER LI (+). The number of elements in such a vector is variable, up to a maximum of 873 (default value is 51.). The vector containing the result can be stored under a letter key, i.e. Y, and can be looked at by typing DISPLAY Y.

Scaling of plots on the OLS is automatic for best fit, or can be controlled. Upon default, however, it is often desirable to look at plots of several functions on a common

*Underlined operators are each a single key-push.

scale. This can be done on the OLS; and the graphs will be overlayed. OLS graphical capabilities are available to users at UCSB on the Culler-Fried terminals, and to Network users using a special graphics socket at UCSB. See NIC 15747, RFC 503 "Socket Number List". For Network users without Culler-Fried keyboards, see NIC 7546, RFC 216 "TELNET Access to UCSB's On-Line System".

B. Symbolic Manipulations Available at MATHLAB

MATHLAB's MACSYMA provides the capability to do many symbolic manipulations in a very straightforward and easy-to-learn manner. Included in these manipulations are:

1) Symbolic integration and differentiation of certain functions.
2) Solutions to equations and systems of equations.
3) Laplace and inverse-Laplace transforms of certain functions.
4) Certain series expansions.
5) Rational simplification of rational functions.

For a more complete description, see "The MACSYMA Users' Manual" by the MATHLAB Group at Project MAC-MIT.

III. A DESCRIPTION OF THE CURRENT IMPLEMENTATION

A variety of programs are used to make up a system to effect this transfer of data.

1) Two functions are defined in Lisp-like language which are loaded into MACSYMA after login in order to facilitate saving a list of expressions to retrieve later to UCSB, and to write this list out to a disk file at MATHLAB for later retrieval.
2) A set of OLS user programs create the batch job which actually performs the retrieval, translation, and storage of these expressions on a specified file on some OLS user directory.
3) The program which actually performs the connection to MATHLAB retrieves the expressions, translates and stores into the OLS is written in PL/1 and exists as a load module on disk at UCSB.

The sequence of operations required in order to retrieve expressions using these various programs is outlined below:

1) The user makes a connection to MIT-MATHLAB in the conventional manner. This can be done either through UCSB-OLS, or through other TELNET programs, or from a TIP.

2) The user logs in at MATHLAB, calls up MACSYMA, and loads the file into the MACSYMA system which facilitates retrieval. (Contains ADDLIST and SAVE functions.)

3) The user performs the desired manipulations at MATHLAB, and saves up a list of line numbers as he goes along using the ADDLIST function. These line numbers represent those expressions he wishes to retrieve. The format for ADDLIST is ADDLIST('<LINE NUMBER>).

4) When the user has completed all the manipulations he wishes to do he saves them on the MIT-MATHLAB disk. (Using SAVE function.) The format for SAVE function is SAVE (<filename 1>). This function writes out, in horizontal form, the list of line numbers in the order the ADDLIST function was invoked to the MIT disk. The filename will be <filename 1>BATCH. SAVE also appends a question mark on the end of the file as an end-of-file indicator.

5) USER disconnect from MATHLAB.

6) User connects to and logs into OLS, and loads a file containing the user programs which produce a virtual job deck for the batch system. A sequence of questions are given to the user by these programs regarding accounting information, and the source file at MIT, and the destination file at at UCSB. The batch job gets submitted automatically, and the transfer and translation is done.

7) After the transfer is completed, the destination file may be loaded into OLS, and the results may be displayed and numerical manipulations can take place.

The form of these user programs, as they are returned is as follows:

LII REAL LOAD (function)

Therefore in order to look at a graph of one of these functions, it is necessary to set up values of various constants, as well as a range of values of the independent variable. It is also

necessary to request a display of the function. This can be done by typing DISPLAY RETURN. It should be noted that the function does exist at the time directly after the user program is called and may be stored under any of the alphabetical keys on the OLS. Storing several of these functions under alphabetical keys will allow them to be called up for plotting on a common scale. For example, if the functions were stored under the keys A, B, and C, they could be displayed on a common scale by typing DISPLAY, ABC RETURN.

IV. LIMITATIONS

A. The program as it stands can only transfer expressions. Equations or functions are not implemented.

B. Variable and constant names at MIT can contain more than one letter, but the current implementation recognizes only one-letter variable names.

C. The program as it stands does not handle complex numbers.

D. The user is subject to failures of three independent systems in order to complete the transfer: the UCSB 360/75, the Network, and the PDP-10 at MIT. This has not proven to be a serious constraint.

E. Software changes at either site can cause difficulties since the programs are written assuming that things won't change. Anyone who has ever had a program that works knows what system changes or intermittent glitches can do to foul things up. With two systems and a Network things are at least four times as difficult. Thanks are due to Jeffrey Golden at PROJECT MAC for helping with ironing things out at MATHLAB, and the UCSB Computer Center for their patience with many I/O bound jobs.

V. POSSIBILITIES FOR ADDITIONS

A. Recognition of complex numbers, possibly for use with LII COMPLEX on the OLS.

B. Addition to translation tables of WMPTALK for recognition of SUM, COSH, SINH, INTEGRATE, DIFF, etc. (Often MATHLAB will not be able to perform an integral or derivative, in which case it will come back with INTEGRATE (Expression) as its answer.)

C. An OLS Utilities package for allowing users to more easily manipulate the numerical vectors describing the expressions, i.e., setting up linear and logarithmic sweeps for the various plots, describing the scale of the plots on the OLS screens.

D. The ability to have an OLS program written from a MATHLAB function, including IF, THEN, ELSE, DO, etc. This would most likely require a more sophisticated parse than is done in the current implementation.

VI. EXAMPLE

An example is included in which a UCSB user:

A. Logs into MATHLAB,

B. Initializes the "SAVE" function,

C. Generates a polynominal function and its derivative and integral,

D. Logs out of MATHLAB,

E. Creates the retrieval job,

F. Waits and then displays the resultant user programs,

G. and, finally, creates the X variable and plots the functions.

Although the sample OLS manipulations are very simple ones it should be noted that the user could compare the retrieved functions with numerical models or even use the functions as subroutines in higher level algorithms. Usage of this combined numeric-symbolic system is limited to the imagination of the user.

The example follows:

Connection to MATHLAB from UCSB
"II LOG MIT-ML RETURN"

USER TELNET
LOGIN TO MIT-ML

MIT MATHLAB PDP-10

ML ITS.796. DDT.514.
9. USERS

```
:LOGIN WMP                              Login to MIT-MATHLAB.

:MACSYMA                                Call up MACSYMA.

THIS IS MACSYMA 212
USE " INSTEAD OF ?
SEE UPDATE > MACSYM;

FIX 212 DSK MACSYM BEING LOADED
LOADING DONE

(C1) BATCH(BATCH,UTILS);                Load BATCH UTILS file.

(UREAD BATCH UTILS DSK WMP) FILE NOT FOUND

(C2) BATCH(BATCH,UTILS,DSK,UCSB);

(C2) LISTX:();
(D2)                                    ()

(C3) ADDLIST(X):=LISTX:CONS(X,LISTX);
(D3)                    ADDLIST(X) := (LISTX : CONS(X, LISTX))

(C4) SAVE(FILENAME):=APPLY(STRINGOUT,APPEND(CONS((FILENAME,BATCH,DSK,UCSB),REVER
SE(LI
STX)),("?")));
(D4) SAVE(FILENAME) :=
     APPLY(STRINGOUT, APPEND(CONS((FILENAME, BATCH, DSK, UCSB), REVERSE(LISTX)),
     (?)))

(D5)                                    BATCH DONE
```

```
(C6) (X**2+3)/(X+1);
                                        2
                                       X  + 3
(D6)                                   ------
                                       X + 1

(C7) INTEGRATE(%,X);

SIN FASL DSK MACSYM BEING LOADED
LOADING DONE
                                         2
                                        X  - 2 X
(D7)                                    -------- + 4 LOG(X + 1)
                                          2

(C8) ADDLIST('D6);
(D8)                                    (D6)

(C9) ADDLIST('D7);
(D9)                                    (D7, D6)

(C10) DIFF(D6,X);
                                                 2
                                        2 X     X  + 3
(D10)                                   ----- - --------
                                        X + 1          2
                                               (X + 1)

(C11) ADDLIST('D10);
(D11)                                   (D10, D7, D6)

(C12) SAVE(MYFILE);

(D12)                                   (D6, D7, D10, ?)

(C13) *******Z          Leave MACSYMA
25156)   .IOT 1,1

:LISTF UCSB

DSK  UCSB
FREE BLOCKS U0 #1 241 U1 #3 345 U2 #5 379
   3   ATTN   BATCH  1  5/23/73 13:53:11
   1   BATCH  UTILS  1  5/23/73 13:11:43
   3   DEMO   WMP    1  5/26/73 15:29:26
   5   DEMO1  BATCH  1  4/29/73 22:41:17
   1   DEMO99 BATCH  1  5/25/73 00:07:15
   5   MYFILE BATCH  1  5/31/73 12:41:50 <-file is in directory at MATHLAB.
   1   _MSGS_ UCSB   0  5/26/73 21:13:53

:LOGOUT
ML ITS 796 CONSOLE 24 FREE. 12:42:35

DISCONNECTION COMPLETE
```

Use ADDLIST function to save line numbers D6 & D7.

Use ADDLIST function to save line number D10.

Write list of lines out to a disk file using SAVE function.

Logout and disconnect.

```
WORK AREAS UPDATED                    Load Retrieval program
LOAD MATHLAB                          "SYST LOAD MATHLAB RETURN"
FILE LOADED

                                      "USER LO ⊕"
RETRIEVE EXPRESSIONS
--------------------

MATHLAB FILE?  (EXP)
-->MYFILE-->MYFILE.                   "MYFILE ENTER"
OLS FILE?  (MYFILE)
-->DEMO11-->DEMO11.                   "DEMO11  ENTER"
OLS FILE
PROTECT CODE?  ()                     "DEMO11  ENTER"
-->DEMO11-->DEMO11.
BATCH JOBNAME?  (MYFILE)              "PARSET ENTER"
-->PARSET-->PARSET.

PRESS ENTER TO SUBMIT JOB             "ENTER"
```

```
VOLUME NEEDED=
JOB SUBMITTED

JOB TO RETRIEVE MATHLAB
EXPRESSIONS IS NOW IN
UCSB-MOD75 BATCH QUEUE.  Some time elapses and batch job is run.
                         Load the retrieved program.
WORK AREAS UPDATED            "SYST LOAD DEMO11 RETURN"
LOAD DEMO11
FILE LOADED

                Display the returned expressions.

(USER LI 0)                   "USER I DISPLAY ⊕ "
LII  REAL  LOAD ((X**2 ⊕ 3)/(X ⊕ 1)):

(USER LI 0)                   "USER I DISPLAY ⊖ "
LII  REAL  LOAD ((X**2 ⊖ 2*X)/2 ⊕ 4* LOG (X ⊕ 1)):

(USER LI 0)                   "USER I DISPLAY ⊛ "
LII  REAL  LOAD (2*X/(X ⊕ 1) ⊖ (X**2 ⊕ 3)/(X ⊕ 1)**2):

USER LI SQ UNDEFINED          "USER I DISPLAY SQ"
```

SAMPLE OLS CURVES GENERATED FOR $-.5 \leq X \leq 4.5$

PROCEDURE

1. Generate $-.5 \leq X \leq 4.5$,

   "II REAL ID ⊕ 1 ⊙ 2.5 ⊖ .5 STORE X"

2. Compute and display curves,

   "USER I ⊕ DISPLAY RETURN" gives (D6)

   "USER I ⊖ DISPLAY RETURN" gives (D7)

   "USER I ⊙ DISPLAY RETURN" gives (D10)